(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 593 062 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **93116655.7**

(51) Int. Cl.5: **G06F 11/00**

(22) Date of filing: **14.10.93**

(30) Priority: **16.10.92 US 961752**

(43) Date of publication of application:
20.04.94 Bulletin 94/16

(84) Designated Contracting States:
AT CH DE FR GB IT LI NL SE

(71) Applicant: **Siemens Industrial Automation, Inc.**
3000 Bill Garland Road
Johnson City, TN 37601(US)

(72) Inventor: **Marks, David Jackson**
515 C Pilgrim Court
Johnson City, TN 37601(US)

(74) Representative: **Fuchs, Franz-Josef, Dr.-Ing. et al**
Postfach 22 13 17
D-80503 München (DE)

(54) Redundant networked database system.

(57) A Redundant Networked Database System is taught. Briefly stated, Control System Computers are designated for primary and backup database operation with applications being inputable to either primary or backup. Upon changes to the database, the primary and backup communication agents communicate with each other and to automatically update the backup. In this fashion, the primary and backup databases are automatically synchronized without manual intervention or the need for reinputting of the changes to the backup database.

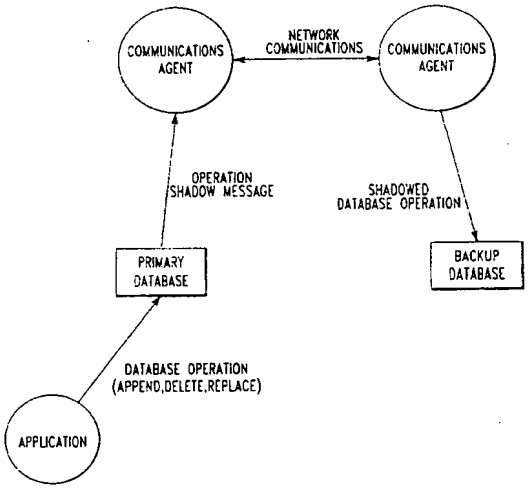


FIG. 1

EP 0 593 062 A2

FIELD OF THE INVENTION

This invention relates, generally, to control systems and more particularly to a control system having automatic database redundancy.

BACKGROUND OF THE INVENTION

It is known that the use of computers in a manufacturing environment is increasing at an ever accelerating rate. Whereas it was previously common to install one or two computers which interacted with remote peripheral sensors and the like, there is an increasing tendency to utilize computers, generally in the form of microprocessors, as close as possible to the events or environment being sampled or controlled.

Accordingly, along with this proliferation of computers there is now the need to network them so as to form a distributed control system. One distinct advantage with distributed processing is the desirability and ability to switch to a backup when, for example, the network or a primary computer goes down or malfunctions.

It is therefore important to have as much information on the secondary computer as the primary so that it can generally perform its function without interruption. Unfortunately, with the increasing use of computers, changes are made to the system on a much more pervasive and frequent basis.

Heretofore, backup databases, those databases which were used in the event of a primary system failure, were run in synchronization with essentially identical equipment running both copies in similar environments. Modifications to one database however, generally had to be made to the other databases and required at least some manual intervention. Accordingly, this required that all modifications had to be done at least twice. This is particularly problematic when complex modifications need to be done which thereby results in tedious duplicative work. Moreover, the chance of an error or discrepancy between the backup and primary databases was greatly increased.

There are a number of schemes which have tried to manage a network type system. One such example may be founded in U.S. Patent No. 5,093,782 "Real Time Event Driven Database Management System" to Muraski et. al., issued March 3, 1992 which attempts to speed up and simplify a database system. However, these systems generally require specialized hardware, unique architecture and the like and are therefore generally not "retrofitable" to existing systems.

Accordingly, it is advantageous and an object of the present invention to keep primary and backup databases synchronized without manual intervention. It is also desirable and yet another object of the present invention to produce a system which allows for hardware or network failure by one portion of the system without affecting the remaining portions.

It is still a further object of the present invention and is also desirable to produce a system which effectively allows for two or more different computers to have effectively identical databases while still allowing for communication between the two.

Still a further object of the present invention is to produce a redundant system which makes the backup database invisible or transparent to the user and which automatically accomplishes synchronization without any special effort on the part of the user.

It is also advantageous and another object of the present invention to produce a redundant network database system, comprising at least two computers, a communication link disposed between them so as to allow communication between the two computers and a means for sensing a change to a database and producing an indication thereof and a communications means for sensing the indication produced by the first means whereby the communication means updates the database of the remaining at least two computers.

DESCRIPTION OF THE DRAWINGS

Reference may be now had to accompany drawings in which:

Figure 1 is a block diagram of the network as envisioned by the present invention; and

Figures 2 and 3 are State diagrams of the Primary and Backup Communications Agent, respectively, of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to Figure 1 there is shown a block or function diagram of the present invention. It is to be understood that in the preferred embodiment of the present invention the within communication database is used in conjunction with a Networked Control System although it is to be understood that other types of networks may be utilized without departing from the spirit and scope of the present invention. Such other types of networks may be, for example, computers on a local area network.

Here there is shown the preferred embodiment using two computers each of which incorporates a Primary Database and a Communications Agent. It is to be understood that although only two computers are shown for ease of understanding and il-

lustration purposes, more computers may be used. Accordingly, a network would encompass substantially more computers although they would operate substantially identical to the manner described below. Shown is the Primary Database having a Communications Agent (as described more fully below) integrated therewith and a Backup Database also having a Communications Agent integrated therewith.

An application source acting on the database is also shown as being operatively connected to the Primary Database, although such sources may be operatively connected to any of the Backup Databases. Such applications may be, for example, in the form of an updated process or process control information which has been provided by external input devices or sensors, recorded process variables or any other information relevant to the operation being monitored or controlled by the network system.

Each database runs on its own computer with the communication agent being a resident database package which is also run on each computer. This communications agent package, as described in fully below, provides communications between the databases. By way of overview, whenever an application program (such as a database shall) cause a modification to the primary database, a message is sent from the database to the local communications agent. The communications agent thereby in turn forwards that message to the remote communication agent, in this case the communication agent resident with the backup database. The remote communication agent thereby upon receipt of this message performs that same operation on the backup database.

In the preferred embodiment of the present invention, communications is in both directions as is normally the case with control system networks. Therefore, should a malfunction occur in the primary database or in the primary database's computer, the backup database can take over until the primary is functioning again. Therefore, the former backup is now in the role of primary and can store all database modifications for later transmittal to the former primary. In this fashion, it is not necessary for applications interacting with the network to be postponed or delayed until the primary system is functioning.

Two kinds of failures are detected and acted on by the present invention; database operation failure and network failure. When a database operation fails on the primary computer, the primary computer's database logs an error. The result is that this shadowing is effectively ceased since the operation failed and was not performed. However, if a database operation should fail during a shadowed transaction (that is, where database exchanges are actually taking place) the transaction is generally rolled back which in turn causes the corresponding transaction on the backup computer to also be rolled back. Accordingly, the failed operation (update, delete or insert) is itself not shadowed.

Should a database operation fail on the backup computer it is automatically logged on the backup. Since, during normal operation it is common to expect that the user is not generally paying attention to the backup, a message is automatically sent to the primary computer so that the failure will also be reported to the primary computer. This therefore allows users or other programs on the primary computer to decide what to do in a predetermined fashion. Depending upon what is desired by the user or the resident programs, it is therefore possible that the two databases will no longer be synchronized. In any event however, the database operation failure will have been at least noted.

The second type of failure, network failure when detected on the primary computer automatically causes the connection between the primary and backup database to be shutdown. The primary computer thereafter waits for a re-connection with the backup. Accordingly, any database operations that come in during this time are queued for later transmission to the backup. Therefore when communications are reestablished, transactions that were in progress are resent, followed by the mentioned queued pending operations.

Should the backup computer detect a failure of the network, the backup computer automatically rolls back any transactions in progress to avoid keeping any files/tables locked on the backup computer. The main reason for this is that when communications is reestablished, it is highly unlikely that any corresponding transactions will remain in progress with the result that there would be no convenient way to tell the backup computer to unlock the files/tables. Thereafter, when communications are reestablished, the backup computer goes back to a waiting state for database shadow messages. This automatically causes the primary computer to resend any transactions that were in progress prior to the failure with the result that synchronization is accomplished.

Referring now to figures 2 and 3 there is shown state flow-chart diagrams of the primary and backup communication agents respectively. It is to be understood that in the preferred embodiment of the present invention, the operations are performed in the sequence and manner as shown although the order of some steps and the like may be changed without departing from the spirit and scope of the present invention.

The primary and backup state diagrams are shown for the communications agent portion of the present invention. It should be noted that the

"START" arrow of each of state diagrams represents start up of the communications agents at boot-up of the control system computers. As can be seen with the primary and backup state diagrams, a connection is initially accomplished between the backup and primary over the network. This connection of course checks for failure conditions. Should a failure condition be detected or exist, the sequence is reinitiated. Additionally, with respect to the primary communication agent shown in figure 2, a failure at any point of actual transmission of the database data results in the re-initiation of a connection to the backup over the network.

As can be seen in figure 2, database operations are sent to the backup database with acknowledgments sent back to the primary database. This type of handshake is readily known and available to one skilled in the art and ensures the existence and integrity of successful communication.

It is recognized that in database systems as envisioned by the present invention, two types of changes to the database are generally accomplished. These changes are in the form of an atomic operation or a transaction operation. A transaction operation is one in which particular files or portions of the database are locked out to external changes unless and until the file is unlocked. This type of operation is generally done when extensive or substantial changes are being made to the database. Conversely, an atomic operation is generally one which is automatic, and relatively minor in breadth. In any event, the success of this transmission is again checked to ensure its integrity.

With respect to figure 3 it can be readily seen how the communication agent receives and processes the information from the primary database. Of particular import is its operation during and just after failure. Accordingly, in the event of communications failure, its previously mentioned roll back of any transactions in progress takes place whereby the new information or data is ignored and the integrity of the database is in effect rolled back to just as it was prior to the failure. In this manner, the backup database can reinitiate or continue from a known correct point, thereby enabling it to operate or act as a primary database for backing up the formerly primary database.

Accordingly, the backup communications are synchronized to the primary without any manual initialization or reinputing of any data. Further, since the communication link is a two-way path, complete bi-directional database synchronization is accomplished.

It is to be understood that the present invention may be modified without departing in spirit and scope thereof and that its breadth not be limited by the specific embodiments but rather only limited by the claims appended hereto.

Claims

1. A redundant network database system, comprising:
   a first computer having a means for storing a primary database and at least one second computer, having a means for storing a backup database;
   Communication link means for connecting and operatively interconnectable with said first computer and said at least one second computer;
   first means disposed in said first computer for sensing a change to said primary database and second means in said at least one second computer for sensing a change in said backup database, said first means and said second means producing an indication thereof of a change in the respective databases; and
   communication agent means disposed in said first computer and in said at least one second computer for sensing the indication produced by said first means and said second means, whereby any changes to said primary database are communicated to said backup database.

2. A device according to claim 1 wherein said communication link is bidirectional such that any changes to said backup database are automatically transmitted to said primary database.

3. A device according to claim 1 wherein communication link failures detected by said first computer are automatically transmitted to said at least one second computer and vice versa.

4. A method for automatic redundant network database management having at least two control system computers, a primary computer and a backup computer each of which has a database, said computers interconnected in a network, comprising the steps of:
   A. Initiating communication between said computers;
   B. Detecting a change in the databases associated with said computers; and
   C. automatically updating the database of each previously unaltered database with the changes made to the said altered database.

5. A method according to claim 4 comprising the additional step of:

D. logging an error condition in the primary computers database upon detecting a failure of a primary database operation.

6. A method according to claim 5 comprising the additional step of:

D. rolling back the change to the primary computer database upon detection of a primary computer database operation.

7. A method according to claim 4 comprising the additional step of:

D. transmitting am error condition from said backup database computer to said primary computer database upon said backup computer detecting a database operation error.

8. A method according to claim 4 comprising the additional step of:

D. said primary computer discontinuing the communication between said computers upon said primary computer detecting a failure in said network.

9. A device according to claim 8 comprising the additional step of:

E. queuing in said primary computer database changes to the primary computer upon failure of the network; and

F. transmitting said queued database information to said backup computer database upon reestablishing of communications in said network between said primary computer and said secondary computer.

10. A method according to claim 4 comprising the additional step of:

D. rolling back any transactions in progress in said backup computer database upon said backup computer database detecting a failure in the network.

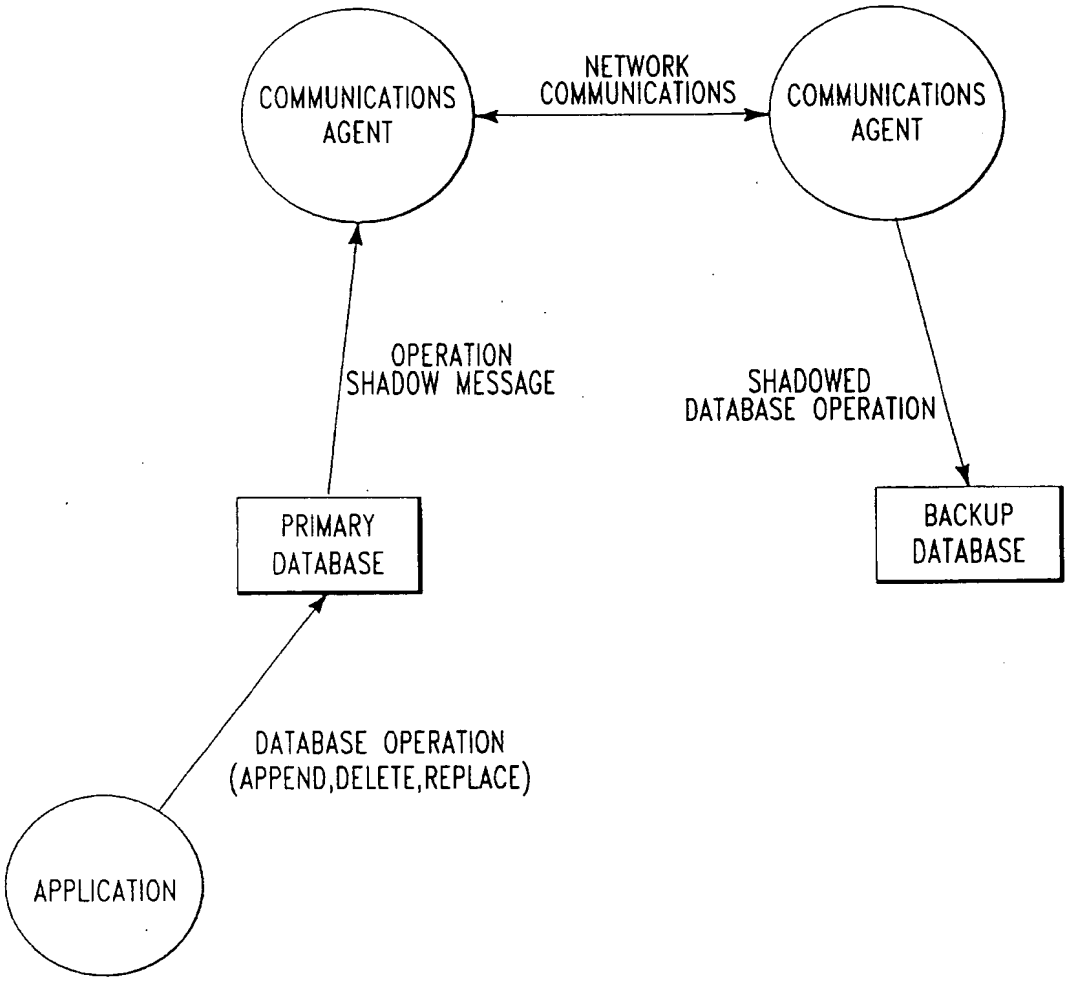
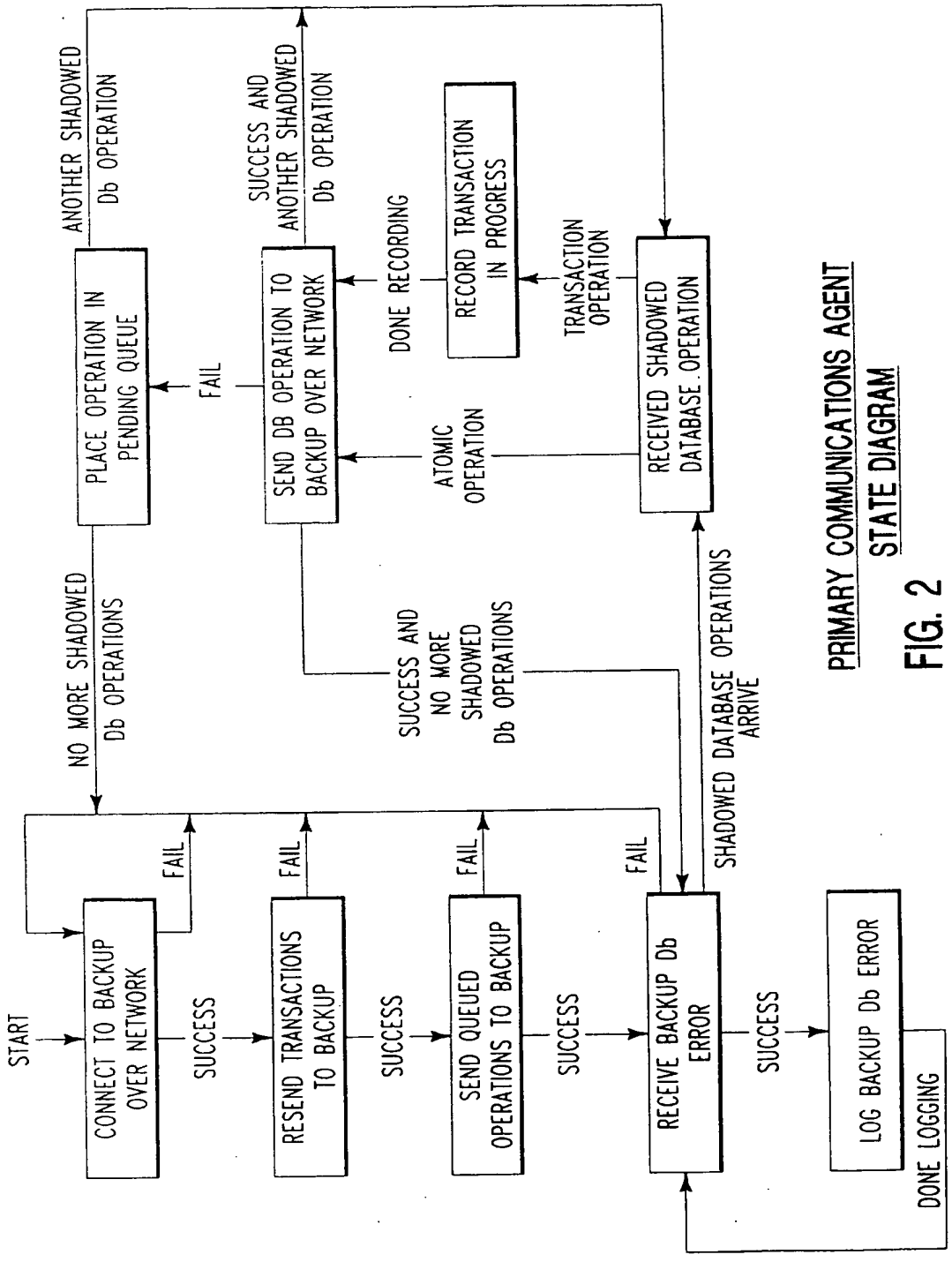
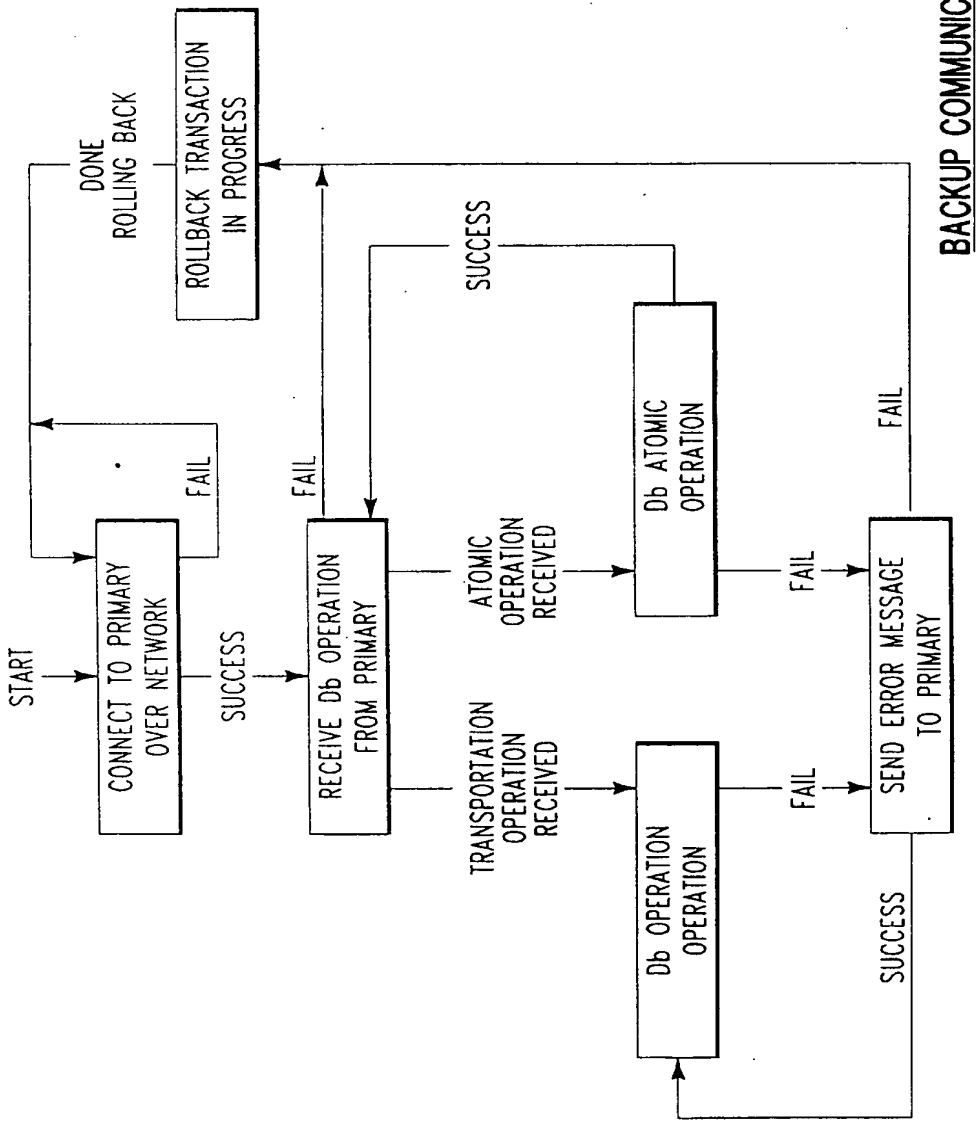

COMMUNICATIONS AGENT
NETWORK COMMUNICATIONS
COMMUNICATIONS AGENT
OPERATION SHADOW MESSAGE
SHADOWED DATABASE OPERATION
PRIMARY DATABASE
BACKUP DATABASE
DATABASE OPERATION (APPEND,DELETE,REPLACE)
APPLICATION

FIG. 1



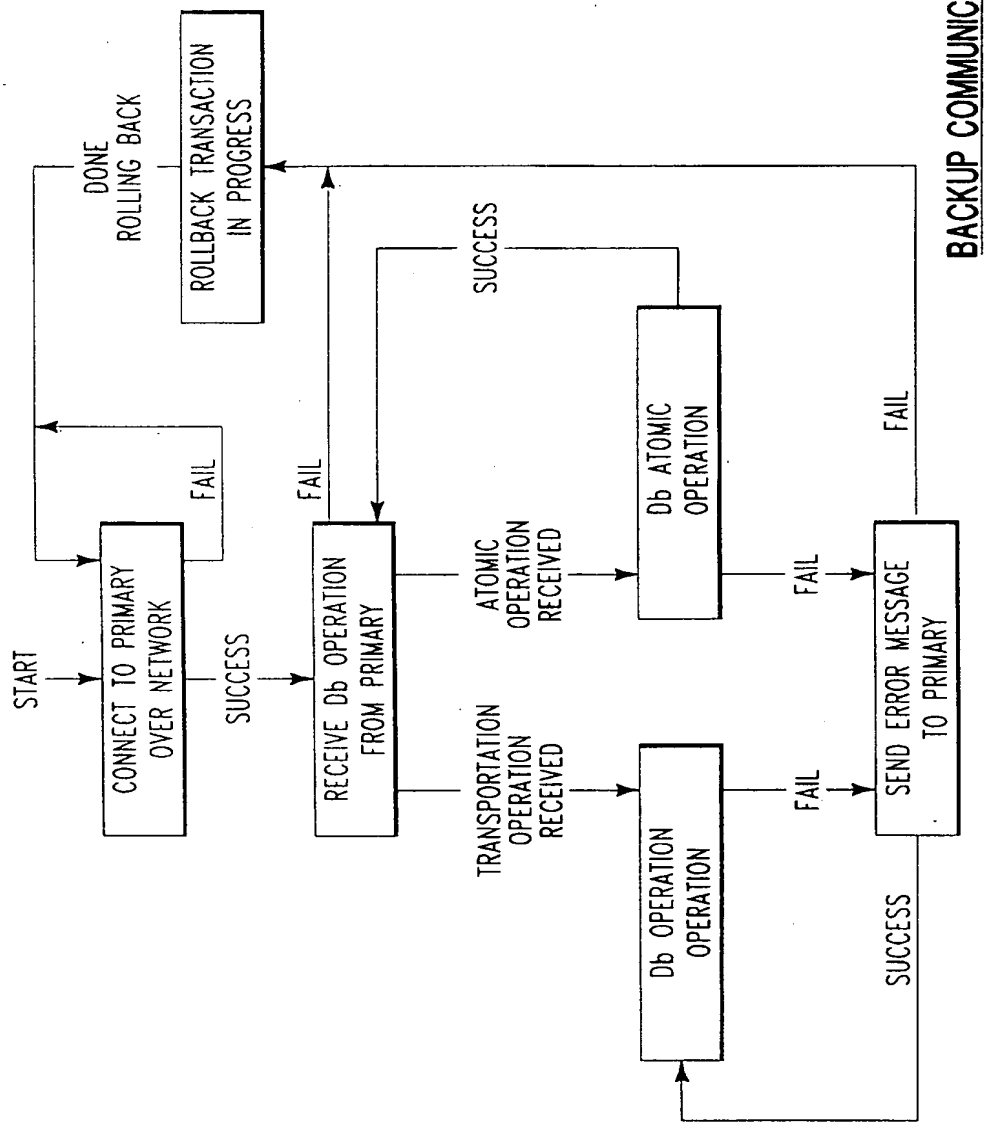
PRIMARY COMMUNICATIONS AGENT
STATE DIAGRAM

FIG. 2



BACKUP COMMUNICATIONS AGENT
STATE DIAGRAM

FIG. 3



BACKUP COMMUNICATIONS AGENT
STATE DIAGRAM

FIG. 3